



**Eiji Adachi Medeiros Barbosa**

**Sistema de Recomendação para  
Código de Tratamento de Exceções**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo programa de Pós-Graduação em Informática do Departamento de Informática da PUC-Rio

Orientador: Prof. Alessandro Fabrício Garcia

Rio de Janeiro

Março de 2012



**Eiji Adachi Medeiros Barbosa**

## **Sistema de Recomendação para Código de Tratamento de Exceções**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Alessandro Fabrício Garcia**  
Orientador

Departamento de Informática – PUC-Rio

**Prof. Renato Fontoura de Gusmão Cerqueira**  
Departamento de Informática – PUC-Rio

**Prof.<sup>a</sup> Christina von Flach Garcia Chavez**  
Departamento de Informática – UFBA

**Prof. José Eugênio Leal**  
Coordenador Setorial do Centro  
Técnico Científico - PUC-Rio

Rio de Janeiro, 30 de março de 2012

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Eiji Adachi Medeiros Barbosa**

Recebeu da Universidade Federal do Rio Grande do Norte (UFRN) o título de bacharel em Ciência da Computação em 2009. Atualmente, participa do grupo de pesquisa OPUS, que funciona junto ao Laboratório de Engenharia de Software da PUC-Rio (LES / PUC-Rio). Seus trabalhos estão relacionados a temas como tolerância a falhas, mecanismos de tratamento de exceções, manutenção e evolução de sistemas de software, métricas e avaliação de software e engenharia de software empírica.

#### Ficha Catalográfica

<p>Barbosa, Eiji Adachi Medeiros</p> <p>Sistema de Recomendação para Código de Tratamento de Exceções / Eiji Adachi Medeiros Barbosa ; orientador: Alessandro Fabrício Garcia. – 2012.</p> <p>125 f. : Il. ; 30 cm</p> <p>Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2012.</p> <p>Inclui bibliografia.</p> <p>1. Informática – Teses. 2. Sistema de recomendação. 3. Tratamento de exceções. 4. Ferramenta de apoio ao desenvolvimento. I. Garcia, Alessandro Fabrício. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.</p>
--

CDD: 004

Para meus pais, Wagner e Emília,  
por todo amor, suporte e confiança.

## **Agradecimentos**

Agradeço primeiramente a meus pais, minhas irmãs e todos familiares. Não há palavras para descrever minha gratidão por todo amor, confiança, incentivo e admiração recebidos. Se eu estou alcançando mais um objetivo em minha vida, é graças a vocês. Vocês são meu alicerce, inspiração e motivação.

Agradeço especialmente a minha avó Nobu, que desde criança me ensina sobre a importância da dedicação aos estudos. Você será para sempre meu exemplo de coragem e perseverança.

Agradeço a Maria Luiza por estar ao meu lado durante todos os momentos deste mestrado. Sua dedicação, disciplina e paciência foram inspiradoras. Seu amor, carinho e compreensão foram a força motriz para que eu alcançasse este objetivo.

Agradeço a todos os amigos de Natal-RN por todo suporte e carinho, mesmo estando ausente. Reencontrá-los nos momentos de folga é sempre uma reinjeção de ânimo para continuar trabalhando.

Agradeço ao meu orientador, Alessandro Garcia, por toda confiança depositada em mim e pela enorme contribuição para o meu crescimento acadêmico. Seu profissionalismo, dedicação e energia para o trabalho são exemplares. Mas agradeço, acima de tudo, pela relação de amizade construída ao longo desses dois anos de mestrado.

Agradeço a todos os amigos de disciplinas cursadas no mestrado pelo companheirismo, encorajamento nos momentos mais difíceis e momentos alegres e descontraídos. Sem a companhia de vocês este caminho teria sido bem mais difícil.

Agradeço a todos os amigos do LES e os integrantes do grupo de pesquisa OPUS. Seus comentários e discussões são sempre enriquecedores. Em especial, agradeço a Francisco Dantas por todo suporte recebido.

Agradeço a todos os professores do Departamento de Informática da PUC-Rio pela contribuição em minha formação acadêmica. Agradeço também a todos os funcionários do departamento por seus serviços prestados.

Agradeço ainda ao CNPq, à FAPERJ e à PUC-Rio pelos auxílios financeiros concedidos, sem os quais este trabalho não teria sido concretizado.

A todos vocês, o meu mais sincero agradecimento.

## Resumo

Barbosa, Eiji Adachi Medeiros; Garcia, Alessandro Fabrício. **Sistema de Recomendação para Código de Tratamento de Exceções**. Rio de Janeiro, 2012. 125 p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Mecanismos de tratamento de exceções são modelos comumente usados em linguagens de programação para a estruturação do fluxo de execução excepcional de módulos de software. Tais mecanismos possibilitam a detecção e a sinalização da ocorrência de exceções, permitindo que módulos de software tomem medidas corretivas que recuperem o sistema para um estado correto. Estudos recentes mostram que falhas recorrentes e problemas de desempenho em aplicações de software estão relacionados à má qualidade do código que implementa o tratamento de exceções. Os estudos preliminares realizados no contexto desta dissertação não apenas corroboram esses resultados, como também evidenciam uma série de outros fatos interessantes que ocorrem durante a evolução de sistemas de software. Em especial, percebeu-se que é bastante comum nas versões preliminares de um sistema a implementação de ações de tratamento de exceções ineficazes. Também se percebeu que pouco (ou nenhum) esforço é realizado em termos de melhorar o tratamento de exceções durante a evolução do sistema de software. Os trabalhos recentes que se propõem a auxiliar desenvolvedores de software a lidar com código de tratamento de exceções falham em auxiliá-los em uma tarefa primordial: implementar código de tratamento de exceções bem estruturado desde as primeiras versões de um sistema. Neste contexto, o presente trabalho propõe um sistema de recomendação capaz de prover exemplos de código tratando exceções de modo a auxiliar desenvolvedores a implementarem seu próprio código em suas aplicações. O sistema extrai informações estruturais do código sob desenvolvimento para buscar por exemplos em um repositório local a fim de recomendar exemplos de código com contexto parecido ao que o desenvolvedor está trabalhando. O sistema implementado neste trabalho recomenda exemplos de código implementados em Java.

## Palavras-chave

Sistema de recomendação; tratamento de exceções; ferramenta de apoio ao desenvolvimento.

## Abstract

Barbosa, Eiji Adachi Medeiros; Garcia, Alessandro Fabrício. **Recommender System for Exception Handling Code**. Rio de Janeiro, 2012. 125 p. MSc. Dissertation – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Exception handling mechanisms are models commonly used in programming languages to structure the exceptional execution flow of software modules. These mechanisms enable the detection and signaling of exception occurrences, allowing these modules to take corrective actions to recover a module to a correct state. Recent studies show that recurrent failures and performance issues in software applications are related to the poor structuring of exception handling code. The preliminary studies performed in the context of this dissertation not only confirm these results, but also show other interesting facts that occur during software evolution. In particular, it was realized that it is often very common to observe ineffective exception handling actions in the first releases of a system. It was also realized that little (or no) effort was allocated to improve the quality of these action during the software evolution. Recent studies that aim aiding software developers to deal with exception handling elements fail in assisting them in a crucial task: implementing well-structured exception handling code since the first release of a system. In this context, this dissertation proposes a recommender system able to provide code fragments implementing exception handling code. The proposed strategy aims to assist developers in implementing their own code in their applications. The system extracts structural information from the code under development to search for examples in a local repository in order to recommend code examples that share similar structural context with the code that the developer is working on. The system implemented in the context of this dissertation recommends code examples implemented in Java.

## Keywords

Recommendation system; exception handling; development aiding tool.

## Sumário

1	Introdução	13
1.1.	Contextualização	13
1.2.	Problemática	14
1.3.	Limitação do estado da arte e estudos preliminares	15
1.4.	Programando tratamento de exceções: um cenário motivador	18
1.5.	Objetivos e visão geral da solução	21
1.6.	Estrutura da dissertação	23
2	Referencial teórico e estudos preliminares	24
2.1.	Tratamento de exceções	25
2.1.1.	Trabalhos relacionados	32
2.1.1.1.	Ferramentas de apoio ao desenvolvedor	32
2.1.1.2.	Boas práticas de desenvolvimento	36
2.2.	Sistemas de recomendação para a engenharia de software	37
2.2.1.	Trabalhos relacionados	40
2.2.1.1.	Auxílio no uso de APIs e <i>frameworks</i>	40
2.2.1.2.	Auxílio ao reuso de código	42
2.3.	Estudos de qualidade de código de tratamento de exceções	44
3	Sistema de recomendação para código de tratamento de exceções	48
3.1.	Restrições e decisões de projeto	49
3.2.	Visão geral	50
3.3.	Mecanismo de coleta de informações	53
3.3.1.	Modelo de dados	53
3.3.2.	Extração	55
3.3.3.	Fonte de exemplos	57
3.4.	Mecanismo de recomendação	60
3.4.1.	Heurísticas de busca	61
3.4.1.1.	Heurística I - Tipo da Exceção Tratada	61
3.4.1.2.	Heurística II - Seqüência de Chamadas	63
3.4.1.3.	Heurística III - Tipos das Variáveis	64

3.4.2. Concretização das heurísticas: consultas e pontuação	66
3.4.2.1. Mecanismo de consultas	66
3.4.2.2. Mecanismo de pontuação	69
4 Implementação	74
4.1. Arquitetura de software	76
4.1.1. Módulo Extrator de Informações	79
4.1.2. Módulo Gerenciador de Repositório	82
4.1.3. Módulo Recomendador	84
5 Avaliação	86
5.1. Avaliação preliminar	86
5.1.1. Avaliação da suficiência dos fatos estruturais	88
5.1.2. Avaliação do ajuste dos pesos	91
5.2. Avaliação de relevância	97
5.2.1. Relevância dos exemplos recomendados	99
5.2.2. Influência do tamanho do conjunto de fatos estruturais	105
5.3. Comparação com trabalhos relacionados	109
5.3.1. Tratamento de exceções	109
5.3.2. Sistemas de recomendação	110
6 Conclusão	114
6.1. Considerações finais	114
6.2. Contribuições	117
6.3. Trabalhos futuros	119
7 Referências Bibliográficas	122

## Lista de Figuras

Figura 1 Sugestão para tratamento de exceções	18
Figura 2 Código de tratamento de exceções gerado automaticamente pelo Eclipse	19
Figura 3 Hierarquia básica de tipos de exceção em Java	29
Figura 4 Evolução de tratadores	44
Figura 5 Evolução de tipos de exceção	46
Figura 6 Modelo de dados	55
Figura 7 Exemplo da Heurística Tipo da Exceção Tratada	62
Figura 8 Exemplo da Heurística Sequência de Chamadas	64
Figura 9 Exemplo da Heurística Tipos das Variáveis	65
Figura 10 Colaboração entre módulos	78
Figura 11 Arquitetura do Módulo de Extração de Informações	80
Figura 12 Arquitetura do Módulo de Gerenciamento de Repositório	82
Figura 13 Arquitetura do Módulo Recomendador	84
Figura 14 Resultado das heurísticas aplicadas aos fatos estruturais <i>As-Is</i>	89
Figura 15 Resultado das heurísticas aplicadas aos fatos estruturais sem propriedades privadas	90

## Lista de Tabelas

Tabela 1 Campos usados pelos termos	57
Tabela 2 Módulos e responsabilidades	77
Tabela 3 Melhores e piores configurações de pesos	93
Tabela 4 Posições da configuração padrão de pesos	94
Tabela 5 10 melhores e 10 piores configurações	95
Tabela 6 Avaliação de relevância	102
Tabela 7 Avaliação de tamanho e proveniência dos fatos estruturais	106
Tabela 8 Comparação de trabalhos relacionados	111